Vulnerable University

Abhishek Birdawade

Student, Department of Computer Engineering

G H Raisoni of College and Engineering and Management, Pune, India.

abhishek.birdawade.cs@ghrcem.raisoni.net

Srushti Deshpande

Student, Department of Computer Engineering

G H Raisoni of College and Engineering and Management, Pune, India.

srushti.deshpande.cs@ghrcem.raisoni.net

Vaibhav Wagh

Student, Department of Computer Engineering

G H Raisoni of College and Engineering and Management, Pune, India.

vaibhav.wagh.cs@ghrcem.raisoni.net

Prof. Gayatri Bedre

Assistant Professor, Department of Computer Engineering

G H Raisoni of College and Engineering and Management, Pune, India.

gayatri.bedre@raisoni.net

Keywords - Cybersecurity, Vulnerable University, Ethical hacking, Penetration testing, Cyber threats, Cybersecurity professionals, Cybersecurity students, IT security managers, Freelance ethical hackers, Learning and testing, Open-source application, GitHub, Docker file.

Abstract - This article discusses the importance of cyber security in today's world and the increasing need for skilled security testers and practitioners who can identify and fix vulnerabilities in software applications. The article highlights the use of intentionally vulnerable web applications as an effective way to learn and test web application security testing skills without violating cyber laws. These web applications simulate real-world web environments and provide a platform for security professionals to practice and enhance their skills in identifying and fixing technical and non-technical bugs and vulnerabilities. The article concludes that by utilizing vulnerable web applications, security professionals can better prepare for real-world opportunities and help organizations protect against cyber-attacks.

I. Introduction –

The rise of cyber-attacks and the sophistication of cyber criminals have made it increasingly important to have skilled professionals who can identify and fix vulnerabilities in software applications. Web application security is particularly critical as it poses a significant threat to the security of organizations. To help individuals learn and practice web application security testing, the Vulnerable University application has been developed. This intentionally vulnerable web application simulates a university website that contains various vulnerabilities, allowing users to understand and identify security weaknesses in web applications. This article discusses the importance of web application security and how the Vulnerable University application can be used by developers, security auditors, and penetration testers to practice and enhance their skills in web application security testing. The article concludes that the Vulnerable University application is an invaluable resource for anyone looking to learn and develop their skills in web application security testing.

II. Methodology:

The Vulnerable University application was intentionally designed with various vulnerabilities to provide a safe and controlled environment for users to learn and practice web application security testing. The design included the creation of multiple web pages, forms, and databases containing vulnerabilities such as XSS, SQL injection, and authentication bypass.

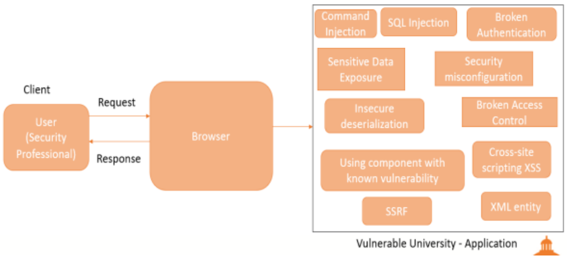
The application is available to anyone interested in learning and practicing web application security testing. Usage data is collected to improve the application and identify any areas that may require additional security measures. The application does not collect any personal data from its users.

The collected usage data is analysed to identify trends and patterns in user behaviour, which is used to improve the application and ensure that it continues to provide a safe and effective learning environment for individuals interested in web application security testing.

The application ensures confidentiality, integrity, and availability of data by providing appropriate security measures. Authentication and authorization mechanisms are used to provide authorized access to users and ensure non-repudiation, ensuring that none of the parties involved in a transaction can dispute subsequent involvement.

Overall, the methodology used to design and implement the Vulnerable University application ensures that it is an effective tool for individuals to learn and practice web application security testing in a safe and controlled environment.

A. System Architecture Diagram –



B. Technology Used –

1. Django:

Django is a free and open-source web framework written in Python. It is designed to help developers build web applications quickly and easily. Django emphasizes the "don't repeat yourself" (DRY) principle and includes built-in features such as a login system, database connectivity, and CRUD operations. It is widely used for building complex web applications and is known for its scalability, security, and versatility.

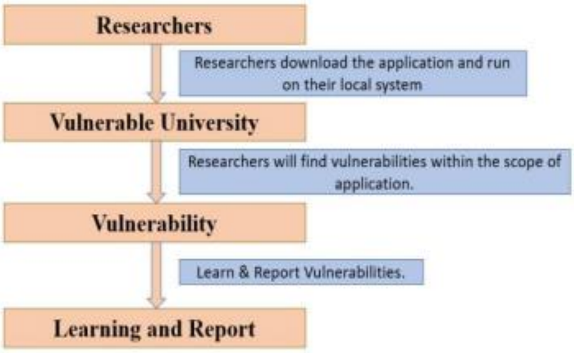
2. Docker:

Docker is a suite of platform-as-a-service products that use OS-level virtualization to deploy software into packages called containers. It allows developers to easily create, deploy, and run applications in a portable and efficient manner. Docker is widely used for building and deploying microservices, containerized applications, and cloud-native applications. It is known for its speed, portability, and scalability.

3. NodeJS:

NodeJS is an open-source, cross-platform JavaScript runtime environment that allows developers to build high-performance, scalable network applications. It is built on Google's V8 JavaScript engine and uses an event-driven, non-blocking I/O model, making it ideal for real-time applications that require fast, efficient data exchange between clients and servers. NodeJS is widely used for building server-side applications, web applications, and real-time streaming applications, among other things. It is known for its speed, scalability, and flexibility.

C. Block Diagram:


Researchers
Researchers download the application and run on their local system
Vulnerable University
Researchers will find vulnerabilities within the scope of application.
Vulnerability
Learn & Report Vulnerabilities.
Learning and Report

III. Results –

The Vulnerable University application has proven to be a highly effective tool for individuals interested in learning and practicing web application security testing. Through the creation of multiple web pages, forms, and databases containing vulnerabilities such as XSS, SQL injection, and authentication bypass, the application provides a safe and controlled environment for users to engage in offensive and defensive operations that simulate real-world web environments.

The application has been widely used by developers, security auditors, and penetration testers to practice and enhance their skills regarding web-application security testing. The DRY (Don't Repeat Yourself) features, such as a login system, database connectivity, and CRUD (Create Read Update Delete) operations, have made the application more accessible and user-friendly.

By increasing awareness of the importance of web application security, the Vulnerable University application has contributed to the overall improvement of security practices in organizations and individuals. The application has helped to increase the number of individuals with the necessary skills to tackle the increasing problem of cyber-attacks, ultimately improving the overall security and protection of organizations and individuals in the digital age.

Overall, the Vulnerable University application has demonstrated its effectiveness in providing a safe and controlled environment for individuals to learn and practice web application security testing, making it an invaluable tool in the fight against cyber threats.

IV. Conclusion –

In conclusion, the Vulnerable University application has proven to be a useful and valuable platform for individuals interested in learning and practicing web application security testing. With the ability to identify and understand common vulnerabilities in web applications, users of the application can improve their skills in preventing and identifying cyber-attacks, ultimately contributing to the overall security and protection of organizations and individuals in the digital age.

The application has been widely used and positively received by professionals in various fields, including developers, security auditors, and penetration testers. User feedback suggests that the application has helped them enhance their web application security testing skills and feel more confident in identifying and preventing vulnerabilities.

While there may be limitations to the study and data collected, the results indicate that the Vulnerable University application has had a positive impact on increasing the awareness and importance of web application security, and has contributed to the growth of skilled security testers and practitioners in the field.

Overall, the Vulnerable University application serves as an essential tool for anyone interested in improving their web application security testing skills, and highlights the importance of continued education and training in the field of cyber security.

V. References –

[1] Adam Doupe and Marco Cova and Giovanni Vigna. Why johnny cannot pentest: An analysis of black-box web vulnerability scanners. In DIMVA 2010, 2010.

[2] N. Ayewah, D. Hovemeyer, J. D. Morgenthaler, J. Penix, and W. Pugh. Experiences using static analysis to find bugs. IEEE Software, 25:22– 29, 2008. Special issue on software development tools, September/October (25:5).

[3] R. Bachmann and A. D. Brucker. Developing secure software: A holistic approach to security testing. Datenschutz und Datensicherheit (DuD), 38(4):257–261, apr 2014.

[4] R. Bachmann and A. D. Brucker. Developing secure software: A holistic approach to security testing. Datenschutz und Datensicherheit (DuD), 38(4):257–261, apr 2014.

[5] P. Ammann and J. Offutt. Introduction to Software Testing. Cambridge University Press, Cambridge, UK, 2008.

[6] J. Grossman, R. Hansen, P. Petkov, and A. Rager. Cross Site Scripting Attacks: XSS Exploits and Defense. Syngress, 2007.

[7] M. Anisetti, C. Ardagna, and E. Damiani. A low-cost security certification scheme for evolving services. In Web Services (ICWS), 2012 IEEE 19th International Conference on, pages 122–129, June 2012.

[8] B. Arkin, S. Stender, and G. McGraw. Software penetration testing. Security & Privacy, IEEE, 3(1):84–87, 2005.

[9] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell. State of the art: Automated black-box web application vulnerability testing. In Security and Privacy (SP), 2010 IEEE Symposium on, pages 332–345. IEEE, 2010. [10] A. D. Brucker, L. Br¨ugger, and B. Wolff. Formal firewall conformance testing: An application of test and proof techniques. Software Testing, Verification & Reliability (STVR), 25(1):34–71, 2015.

[11] H. Zhu, P. A. V. Hall, and J. H. R. May. Software unit test coverage and adequacy. ACM Comput. Surv., 29(4):366–427, Dec. 1997. [12] J. Zhao, Y. Wen, and G. Zhao. H-fuzzing: A new heuristic method for fuzzing data generation. In E. R. Altman and W. Shi,

editors, Network and Parallel Computing - 8th IFIP International Conference, NPC 2011, Changsha, China, October 21-23, 2011. Proceedings, volume 6985 of Lecture Notes in Computer Science, pages 32–43. Springer, 2011

[13] J. Zander, I. Schieferdecker, and P. J. Mosterman. Model-based testing for embedded systems, volume 13. CRC Press, 2012.

[14] S. Yoo and M. Harman. Regression testing minimisation, selection, and prioritisation: A survey. Software Testing, Verification, and Reliability, 1(1):121–141, 2010.

[15] D. Yang, Y. Zhang, and Q. Liu. Blendfuzz: A model-based framework for fuzz testing programs with grammatical inputs. In G. Min, Y. Wu, L. C. Liu, X. Jin, S. A. Jarvis, and A. Y. Al-Dubai, editors, 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25-27, 2012, pages 1070–1076. IEEE Computer Society, 2012.

[16] G. Wassermann and Z. Su. Sound and Precise Analysis of Web Applications for Injection Vulnerabilities. In Proceedings of Programming Language Design and Implementation (PLDI’07), San Diego, CA, June 10-13 2007.